



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,042	11/21/2003	Jacob Strom	8196-16	2544
20792	7590	05/18/2006	EXAMINER	
MYERS BIGEL SIBLEY & SAJOVEC			BROOME, SAID A	
PO BOX 37428				
RALEIGH, NC 27627			ART UNIT	PAPER NUMBER
			2628	

DATE MAILED: 05/18/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

<b>Office Action Summary</b>	<b>Application No.</b>	<b>Applicant(s)</b>
	10/720,042	STROM ET AL.
	<b>Examiner</b>	<b>Art Unit</b>
	Said Broome	2628

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --  
**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) Responsive to communication(s) filed on 23 February 2006.
- 2a) This action is FINAL.                                    2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) Claim(s) 1-40 is/are pending in the application.
  - 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) Claim(s) \_\_\_\_\_ is/are allowed.
- 6) Claim(s) 1-40 is/are rejected.
- 7) Claim(s) \_\_\_\_\_ is/are objected to.
- 8) Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on \_\_\_\_\_ is/are: a) accepted or b) objected to by the Examiner.
 

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
  - a) All    b) Some \* c) None of:
    1. Certified copies of the priority documents have been received.
    2. Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
    3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) Notice of Informal Patent Application (PTO-152)
- 6) Other: \_\_\_\_\_

**DETAILED ACTION**

***Response to Amendment***

1. This office action is in response to an amendment filed 2/23/2006.
2. Claims 1-28 are original.
3. Claims 29-40 have been added by the applicant.

***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Claims 1, 3, 6, 17, 23, 25 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Olsen et al.(US Patent 6,094,2000) in view of Wood et al.(US Patent 6,204,856).

Olsen et al. teaches all the limitations of claims 1 and 23 except for defining a plurality of rows of tiles comprising a plurality of rows of pixels with each tile including at least two rows of pixels. Regarding the preamble of claim 1, Olsen teaches a graphics processing method in column 2 lines 34-37, and is also illustrated in Figure 4 as element 40. Regarding the preamble of claim 23, Olsen et al. teaches a computer graphics system that determines occlusion for graphics primitives, which would contain computer program code embodied on a computer readable medium in order to execute the occlusion computations. Regarding claims 1 and 23,

Olsen et al. teaches setting occlusion flags for respective tiles of a row of tiles, or pixels comprising the region for a graphics primitive, and whether the respective representative depth values meet an occlusion criterion in column 5 lines 52-56 where it is described that the depth values of the region comprising the pixels from the graphics primitive are compared and a determination of occlusion is then determined based on the result from the depth compare result register, illustrated in Figure 3 as element 35. Olsen et al. also teaches processing pixels in rows of pixels in a row-by-row manner responsive to the occlusion flags in column 5 lines 22-23 where it is described that the depth values of each pixel of the graphics primitive, which contains rows of tiles, are processed, therefore each row would be processed as each pixel comprising the rows would be processed. Again Olsen et al. fails to teach defining a plurality of rows of tiles comprising a plurality of rows of pixels with each tile including at least two rows of pixels. Wood et al. teaches a defined plurality of blocks of tiles comprising rows of pixels in column 1 lines 6-10, in which each tile includes pixels from at least two rows of pixels, as described in column 4 lines 42-44 and is also illustrated in Figure 2. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al. with Wood et al. because this combination would provide the determination of occlusion of each row of pixels comprising the tiles of a graphics primitive, which would reduce computation time by providing only the visible pixels to be rendered.

Regarding claim 3, 17 and 25, Olsen et al. fails to teach processing rows of pixels using a zig-zag traversal algorithm. Wood et al. teaches processing pixels contained in the rows of tiles using a zig-zag traversal algorithm in column 5 lines 15-20. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al. with Wood et al. because this

combination would provide an efficient algorithm that accurately processing the rows of pixels within a tile.

Regarding claim 6, Olsen et al. teaches a color buffer that stores respective color values in column 2 lines 62-64. Olsen et al. also teaches storing a color values and a depth value for the graphics primitive for the pixel in the color buffer and the depth buffer respectively in column 2 lines 62-64.

Regarding claim 26, Regarding the preamble of claim 26, Olsen et al. teaches a computer graphics system that determines occlusion for graphics primitives, which would contain computer program code embodied on a computer readable medium in order to execute the occlusion computations. Wood et al. teaches a defined plurality of blocks of tiles comprising rows of pixels in column 1 lines 6-10, in which each tile includes pixels from at least two rows of pixels, as described in column 4 lines 42-44 and is also illustrated in Figure 2. Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels which are known to comprise a tile of pixels. Olsen et al. also teaches detecting a flag, or logic value, indicating non-occlusion in column 6 lines 8-13. Olsen et al. also describes processing a pixel for the graphics primitive in steps 42 and 43 of Figure 4, where it is illustrated that the pixels defining the graphics primitive are processed.

Claims 2, 4, 5, 7-16, 18, 20-22, 24, 27-29, 32-34 and 36-40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Olsen et al. in view of Wood et al. in further view of Larson (US Patent 6,313,839).

Regarding claim 2, Olsen et al. teaches processing pixels responsive to occlusion flag, which is equivalent to the depth compare result register, in column 6 lines 7-12. Olsen et al. and Wood et al. fail to teach a first tile, or region, and a second tile for processing. Larson et al. teaches a first and second region, or tile, for processing in column 6 lines 8-10. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide faster processing of occlusion detection for multiple tiles, which would enable quicker rendering of the visible pixels of graphics primitives.

Regarding claims 4 and 22, Olsen et al. teaches in column 6 lines 8-13, setting an occlusion flag, or depth compare result register, for respective pixels. Olsen and Wood et al. fail to teach occlusion flags and respective occlusion threshold depth values stored in a tile occlusion information cache, determining a maximum depth value and comparing the maximum depth value to the cached occlusion threshold depth value. Larson teaches storing occlusion threshold depth values in a cache memory in column 5 lines 63-66, and determining a maximum depth value for the graphics primitive in column 1 lines 59-60. Larson also teaches comparing the maximum depth value with the cached occlusion threshold depth value in column 2 lines 46-51. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide decreased processing time

of pixel of the graphics primitive during determination of occlusion by storing occlusion flags in a cache memory.

Regarding claim 5, Olsen et al. and Wood et al. teach all the limitations except a depth buffer that stores respective occlusion threshold depth values. Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels which are known to comprise a tile of pixels. Olsen et al. also teaches detecting a flag, or logic value, indicating non-occlusion in column 6 lines 8-13. Olsen et al. also describes processing a pixel for the graphics primitive in the tile without retrieving an occlusion threshold depth value for the pixel from the depth buffer in steps 42 and 43 of Figure 4, where it is illustrated that the pixels defining the graphics primitive are processed without referencing an occlusion threshold depth value. Again, Olsen et al. and Wood et al. fail to teach a depth buffer that stores respective occlusion threshold depth values. Larsson teaches a z buffer, or depth buffer, which stores the occlusion threshold depth values, or maximum and minimum z values, in column 1 lines 55-57. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient access to occlusion threshold values by storing them into a depth buffer.

Regarding claim 7, Olsen et al. and Wood et al. teach all the limitations except for comparing the determined depth value to the occlusion threshold depth value and updating the occlusion threshold depth value in the tile occlusion information threshold cache to the determined depth value. Olsen et al. teaches determining a depth value for the graphics primitive for the pixel in column 3 lines 53-57 where it is described that pixel data is determined including

the depth value. Again, Olsen et al. fails to teach comparing the determined depth value to the occlusion threshold depth value and updating the occlusion threshold depth value in the tile occlusion information threshold cache to the determined depth value. Larson teaches comparing the determined depth value to the occlusion threshold depth value in the tile occlusion information cache in column 2 lines 47-51, where it is described that the z value, or determined depth value, is compared to the maximum z value, or occlusion threshold depth value, that is stored in the cache memory. Larson also teaches updating the occlusion threshold depth value in response to the comparison to the determined depth value in column 2 lines 52-61, where it is described that the z value stored in the cache memory is updated and replaced by the determined depth value that was compared. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 4.

Regarding claim 8, Olsen et al. and Wood et al. fail to teach the claimed limitations. Larson teaches an aggregate tile occlusion information memory configured to store respective occlusion threshold depth values in column 4 lines 30-34, where it is described that a memory area stores the minimum and maximum depth values, or occlusion threshold depth values, within each region, which is equivalent to a tile comprising a block of pixels consisting of rows. Larson also teaches reading out z max and z min, which are occlusion threshold depth values, from the z limit buffer, which is stored in memory, and storing them into the cache memory. Larson teaches updating the z min, or occlusion threshold depth value, in the cache memory in column 6 lines 23-33. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide

efficient storage of depth values for a region, which would then be utilized to determine and to responsively update occlusion properties for pixels within that region.

Regarding claim 9, Olsen et al. and Wood et al. teach all the limitations except detecting that the tile, or region of rows of pixels, has an occlusion flag indicating possible occlusion and updating the occlusion threshold depth value for the tile in the tile occlusion information cache responsive to the written z-value of the pixel. Olsen et al. teaches setting an occlusion flag, or depth compare result register, to indicate possible occlusion in column 6 lines 8-13. Again, Olsen et al. and Wood et al. fail to teach detecting that the tile, or region of rows of pixels, has an occlusion flag indicating possible occlusion and updating the occlusion threshold depth value for the tile in the tile occlusion information cache responsive to the written z-value of the pixel.

Larson teaches updating the occlusion threshold depth value, or min and max z value, in the cache memory responsive to the written z-value of the pixel in column 3 lines 9-18, where it is described that the min and max z values are updated and the received z value is written to a location in the cache memory. Larson describes comparing a depth value to an occlusion threshold depth value for the pixel in the depth buffer responsive to the determination of occlusion in column 4 lines 54-66. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide a reduction of processing time used to store and compare z values of row of pixels by storing the occlusion flags in a cache memory thereby ensuring quick and accurate determination of occluded regions.

Regarding claim 10, Olsen et al. and Wood et al. teach all the limitations except updating the occlusion threshold depth value for the tile in the tile occlusion information cache responsive

to the written z-value of the pixel. Olsen et al. teaches a color buffer that stores respective color values in column 2 lines 62-64. Olsen et al. also teaches storing a color values and a depth value for the graphics primitive for the pixel in the color buffer and the depth buffer respectively in column 2 lines 62-64. Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels which are known to comprise a tile of pixels. Again, Olsen et al. and Wood et al. fail to teach updating the occlusion threshold depth value for the tile in the tile occlusion information cache responsive to the written z-value of the pixel. Larson teaches updating the occlusion threshold depth value, or min and max z value, in the cache memory responsive to the written z-value of the pixel in column 3 lines 9-18, where it is described that the min and max z values are updated and the received z value is written to a location in the cache memory. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient access to occlusion threshold values and color values by storing them into a depth and color buffers respectively.

Regarding claims 11 and 13, Olsen et al. and Wood et al. teach all the limitations except determining whether a second row of pixels is in a first row of tiles. Regarding the preamble of claims 11 and 13, though Larson does not explicitly teach storing occlusion flags in a cache memory, the reference does teach storing z values in a cache memory in column 2 lines 47-51 where it is described that a determination of whether the tested z value is occluded, therefore it would be obvious to store the result of the determination as an occlusion flag in the cache as well. Regarding claims 11 and 13, Larson teaches processing specific regions, or tiles, of pixels

in column 6 lines 8-10, therefore it would be a minor computation to process the first row of pixels specifically, which contain z values stored in a cache memory, as described in column 2 lines 47-51. Though Larson does not explicitly teach storing occlusion and status flags, the reference does teach the determination of occlusion for z values in the region in column 2 lines 47-51, therefore it would be obvious to indicate occlusion based on the determined result using a flag or other indicator. Regarding claims 11 and 13, Larson teaches separately processing different regions, or tiles, of pixels to test the z values for occlusion in column 6 lines 5-10, therefore it would have been an obvious modification to determine whether a pixels from a second row is within a first row of tiles. Regarding claim 11, Larson also teaches independently processing a first and second region, or tile in column 6 lines 8-10 which contain all the rows of pixels, using data from the cache memory in column 6 lines 5-18, therefore it would have been a minor modification to obtain information from a first row of pixels if the second row of pixels is contained the first row of tiles. Regarding claim 13, Larson teaches independently processing regions, or tiles, in column 6 lines 8-10 which contain z values that are tested to determine occlusion and stored in a cache memory as described in column 2 lines 47-51, therefore it would be an obvious modification to write, load and process any depth value responsive to the region and to store the values into the memory comprising tile information, as described in column 4 lines 29-34, or the cache memory. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide a reduction of processing time used to store and compare z values of row of pixels by storing the occlusion flags in a cache memory thereby ensuring timely and accurate determination of occluded regions.

Regarding claims 12 and 13, Olsen et al. and Wood et al. teach all the limitations except establishing an aggregate tile occlusion information memory to store respective occlusion threshold depth values for all tiles and storing occlusion threshold depth values for the first row of tiles from the aggregate tile occlusion information memory in the tile occlusion information cache. Olsen et al. teaches setting the occlusion and status flags in the memory to predetermined values in column 4 lines 63-64, where it is described that when the depth compare result register, which is equivalent to the occlusion flag, is initialized in step 41 of Figure 4 is has a predetermined logic value, or status flag. Again, Olsen et al. and Wood et al. fail to teach establishing an aggregate tile occlusion information memory to store respective occlusion threshold depth values for all tiles and storing occlusion threshold depth values for the first row of tiles from the aggregate tile occlusion information memory in the tile occlusion information cache. Regarding claims 12 and 13, Larson teaches an aggregate tile occlusion information memory configured to store respective occlusion threshold depth values in column 4 lines 30-34, where it is described that a memory area stores the minimum and maximum depth values, or occlusion threshold depth values, within each region, which is tile comprising a block of pixels consisting of rows. Larson also teaches storing occlusion threshold depth values, or min and max z values, from a first region(column 6 lines 8-9), from the aggregate tile occlusion information memory, or RAM as describe din column 4 lines 30-34, into the tile occlusion information cache, or cache memory as described in column 6 lines 23-30. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 11.

Regarding claim 14, Olsen et al. and Wood et al. fail to teach the limitations. Larson teaches separately processing different regions, or tiles, of pixels to test the z values for occlusion in column 6 lines 5-10, therefore it would have been an obvious modification to determine particular locations of rows and whether a pixels from a second row is within a first row of tiles. Larson also teaches updating the occlusion threshold depth value in response to the comparison to the determined depth value in column 2 lines 52-61, where it is described that the z value stored in the cache memory is updated and replaced by the determined depth value that was compared. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient storage of depth values for a region, which would then be utilized to determine and to responsively update occlusion properties for pixels within that region.

Regarding claim 15, Olsen et al. and Wood et al. teach all the limitations except a display and a graphics processor coupled to the display. Olsen et al. teaches setting an occlusion flag, or depth compare result register, to indicate possible occlusion in column 6 lines 8-13, where it is described that the register is set to a certain value indicating the visibility of the pixel under analysis. Olsen et al. also teaches processing pixels in rows of pixels in a row-by-row manner responsive to the occlusion flags in column 5 lines 22-23 where it is described that the depth values of each pixel of the graphics primitive, which contains rows of tiles, are processed, therefore each row would be processed as each pixel comprising the rows would be processed as well. Wood et al. describes a plurality of rows of tiles in a graphics display in column 4 lines 41-44. Wood et al. also describes each tile including pixels from at least two rows of pixels in column 4 line 44 where it is described that a tile could contain 8x8 pixels and therefore, based on

those dimensions, contains at least two rows of pixels. Again, Olsen et al. and Wood et al. fail to teach a display and a graphics processor coupled to the display. Regarding the preamble of claim 15, Larson teaches an apparatus in column 1 lines 52-54. Regarding claim 15, Larson also teaches a display in column 3 lines 46-48, and it is also illustrated in Figure 1 as element 21. Larson also teaches a graphics processor coupled to the display in column 3 lines 57-59. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide an apparatus that determines occlusion of each row of pixels comprising the tiles for a graphics primitive, which would reduce computation time by providing the visible pixels to be rendered.

Regarding claim 16, Olsen et al. teaches processing a portion of pixels in a row of tiles, or region, responsive to the occlusion flag, or depth compare result register, in column 6 lines 7-12. Olsen et al. and Wood et al. fail to teach the graphics processor and a first and second tile for processing. Larson teaches a graphics processor in column 3 lines 57-59. Larson also describes a first and second region, or tile, in column 6 lines 8-10. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 15.

Regarding claim 18, Olsen et al. and Wood et al. fail to teach the claimed limitations. Larson teaches determining a maximum depth value for the graphics primitive in column 1 lines 59-60. Larson also teaches that a depth value, or z value, is compared to determine if that value is less than an established non occluded minimum depth value in column 6 lines 23-30, therefore it would have been a minor modification to compare the maximum depth value with the

minimum non occlusion depth value. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 15.

Regarding claim 20, Olsen et al. and Wood et al. teach all the limitations except a display and a graphics processor coupled to the display. Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels which are known to comprise a tile of pixels. Olsen et al. also teaches detecting a flag, or logic value, indicating non-occlusion in column 6 lines 8-13. Olsen et al. also teaches processing a pixel for a graphics primitive in column 5 lines 22-23. Again, Olsen et al. and Wood et al. fail to teach a display and a graphics processor coupled to the display. Regarding the preamble of claim 20, Larson teaches an apparatus in column 1 lines 52-54. Larson also teaches a display in column 3 lines 46-48, and it is also illustrated in Figure 1 as element 21. Larson also teaches a graphics processor coupled to the display in column 3 lines 57-59. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 15.

Regarding claim 21, Olsen et al. teaches processing a pixel for the graphics primitive in the tile without retrieving an occlusion threshold depth value for the pixel from the depth buffer in steps 42 and 43 of Figure 4, where it is illustrated that the pixels defining the graphics primitive are processed without referencing an occlusion threshold depth value. Olsen et al. and Wood et al. fail to teach a graphics processor that maintains a depth buffer configured to store respective occlusion threshold depth values. Larson teaches a graphic processor in column 3 lines 57-59, which contains a z buffer, or depth buffer, which stores the occlusion threshold depth values in column 4 lines 35-39. It would have been obvious to one of ordinary skill in the

art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide efficient access to occlusion threshold values by storing them into a depth buffer.

Regarding claim 27, Olsen et al. also describes processing a pixel for the graphics primitive in the tile without retrieving an occlusion threshold depth value for the pixel from the depth buffer in steps 42 and 43 of Figure 4, where it is illustrated that the pixels defining the graphics primitive are processed without referencing an occlusion threshold depth value. Olsen et al. and Wood et al. fail to teach a depth buffer that stores respective occlusion threshold depth values. Larson teaches a z buffer, or depth buffer, which stores the occlusion threshold depth values, or maximum and minimum z values, in column 1 lines 55-57. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 21.

Regarding claim 29, Olsen et al. teaches all the limitations except defining a plurality of tiles in a graphics display field, each tile comprising a plurality of pixels and determining a maximum depth value for a graphic primitive. Olsen et al. teaches setting an occlusion flag in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels. Olsen et al. also describes processing a pixel for the graphics primitive in steps 48 of Figure 4, where it is illustrated that the pixels defining the graphics primitive are processed in response to the occlusion flag result of step 47. Again, Olsen et al. fails to teach defining a plurality of tiles in a graphics display field, each tile comprising a plurality of pixels, setting an occlusion flag for a tile responsive to a comparison of a previously determined minimum depth value for the tile to the maximum depth value for the graphics primitive and determining a maximum depth value for a graphic primitive. Wood et al. teaches

defining a plurality of tiles in a graphics display field, each tile comprising a plurality of pixels in column 1 lines 6-10, in which each tile includes pixels, as described in column 4 lines 42-44 and is also illustrated in Figure 2. Olsen et al. and Wood fail to teach determining a maximum depth value for a graphic primitive. Larson teaches that a depth value, or z value, is compared to determine if that value is less than an established non occluded minimum depth value in column 6 lines 23-30, therefore it would have been a minor modification to compare the determined maximum depth value with the minimum non occlusion depth value. Larson also teaches determining a maximum depth value for a graphic primitive in column 1 lines 59-60 (“The maximum Z value corresponds to the largest Z value of a region of Z values.”), where the z value is the maximum depth value, as known in the art and described in column 1 lines 19-25 (“Z buffer depth comparison tests are used to determine whether a new Z coordinate received in the frame buffer controller corresponds to a pixel that will be visible when displayed...”). The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 11.

Regarding claim 32, Olsen et al. fails to teach the limitations. Wood et al. teaches a defined plurality of blocks of tiles comprising rows of pixels in column 1 lines 6-10. However, Olsen et al. and Wood et al. fail to teach a maximum depth within that tile. Larson teaches the maximum depth value represents a depth greater than or equal to all possible depth values that the graphics primitive may have in a tile in column 1 lines 59-60 (“The maximum Z value corresponds to the largest Z value of a region of Z values.”) and in column 4 lines 40-45 (“...to perform the depth comparison test for the pixels of the region.”). The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 29.

Regarding claim 33, Olsen et al. and Wood et al. fail to teach the limitations. Larson teaches determining a maximum depth value, in column 1 lines 59-60 (“The maximum Z value corresponds to the largest Z value of a region of Z values.”), comprises determining a maximum depth of vertices of the graphics primitive in column 2 lines 10-13 (“Each primitive has Z values associated with each vertex of the primitive. Each received Z value is associated with a region of Z values in the Z buffer memory device”). The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 11.

Regarding claim 34, Olsen et al. and Wood et al. fail to teach the limitations. Larson teaches determining a maximum depth value comprising determining a maximum depth of a plane or image of the graphics primitive in the tile in column 1 lines 31-40 (“In these types of systems, Z buffer depth comparison tests are performed by reading the old Z coordinate for the pixel from the Z buffer...writing the new Z coordinate and the associated pixel color into the Z buffer and image buffer...”) and in column 1 lines 59-60 (“The maximum Z value corresponds to the largest Z value of a region of Z values.”). The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 11.

Regarding claim 36, Olsen teaches all the limitations except a first tile in a row of tiles. Olsen et al. also teaches processing pixels in rows of pixels in a row-by-row manner responsive to the occlusion flags in column 5 lines 22-23 where it is described that the depth values of each pixel of the graphics primitive are processed, therefore each row would be processed as each pixel comprising the rows would be processed as well. Again, Olsen et al. fails to teach a first tile in row of tiles. Wood et al. describes a plurality of rows of tiles in a graphics display in column 4 lines 41-44 (“An image is rendered by sequentially and independently rendering small

areas of the screen (tiles)...a tile has dimensions of 8.times.8 pixels.“), therefore these are several areas or tiles with the image. Wood et al. also describes each tile including pixels from at least two rows of pixels in column 4 line 44 where it is described that a tile could contain 8x8 pixels and therefore the tile contains at least two rows of pixels. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 11.

Regarding claim 37, Olsen et al. fails to teach the limitations. Wood et al. teaches processing pixels for the graphics primitive in a row-by-row fashion comprises processing the graphics primitive using a zig-zag traversal algorithm to teach defining tiles comprising rows of pixels in column 1 lines 4-10 (“...a method and apparatus for handling texture and other attributes in rendering of graphic images composed of a plurality of triangular image primitives...in a tile-based renderer, where those images are generated as a series of regular pixel blocks (tiles)...“) and in column 5 lines 15-23 (“...reading the tile fragments in zig-zag raster scanning order, starting at top left, scanning even numbered rows from left to right, odd numbered rows from right to left...“). The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 11.

Regarding claim 38, Olsen et al. fails to teach the limitations. Wood et al. teaches processing tiles in column 4 lines 41-44 (“An image is rendered by sequentially and independently rendering small areas of the screen (tiles)...a tile has dimensions of 8.times.8 pixels.“), therefore these are several areas or tiles with the image. However, Olsen et al. and Wood et al. fail to teach processing portions of tiles for a graphics primitive. Larson teaches processing more than one region or portions z values in column 7 lines 16-19 (“...the Z values for one or more regions and the other is utilized for storing the Z MAX and Z MIN values for the

regions.“) and in column 6 lines 5-10 (“all of the Z values for a particular region do not have to be tested in the frame buffer controller 40 before a Z value associated with a different region can be tested...a Z value corresponding to one region can be tested and then a Z value corresponding to a second region can be tested.“), therefore it would be possible to process any portion or region of a first or second tile for the graphics primitive. The motivation to combine the teachings of Olsen et al., Wood et al. and Larson is equivalent to the motivation of claim 11.

Regarding claim 39, Olsen et al. teaches a graphics processor in column 2 lines 42-45 (“The central processor provides the graphics primitives that are to be rendered and manages certain operations of the occlusion culling circuit.“).

Regarding claim 40, Olsen et al. teaches a computer program product or software comprising program code, or set of instructions, embodied in a computer-readable medium as described in column 4 lines 28-30 (“The circuit 30 can be operated, for example, in an OpenGL.RTM. interface whereby software commands are used...“) and in column 4 lines 45-53 (“...the invention is not limited to this particular software implementation but, as will be apparent to those skilled in the art, may be implemented in other software or in hardware. The occlusion culling circuit 30 is preferably implemented in a combination of hardware and software.“), where the software is described to be executed or implemented on a computer system.

Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Olsen et al.(US Patent 6,094,200) in view of Wood et al.(US Patent 6,204,856), in further view of Larson(US Patent 6,313,839) and in further view of Dye et al.(US Patent 6,518,965).

Regarding claim 19, Dye et al. teaches a display and graphics processor housed in a portable electronic device in column 6 lines 32-46. It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al, Larson and Dye et al. because this combination would provide a more efficient use of the limited resources contained in portable electronic devices.

Claims 30, 31 and 35 are is rejected under 35 U.S.C. 103(a) as being unpatentable over Olsen et al. in view of Wood et al., in further view of Larson and in further view of Duluk, Jr. et al. (US Patent 6,476,807).

Regarding claim 30, Olsen et al. teaches setting an occlusion flag in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels. However, Olsen et al., Wood et al. and Larson fail to teach a comparison of a previously-determined minimum depth value for the tile to the maximum depth value for the graphics primitive comprises setting the occlusion flag to indicate that the graphics primitive is not occluded in the tile responsive to the minimum depth value for the tile exceeding the maximum depth value for the graphics primitive. Duluk, Jr. et al. teaches performing a comparison within a stamp region, or region that comprises a tile as described in column 20 lines 44-52 (“...rectangular grid of pixels, and the window is divided into tiles... further divided into stamps ...stamps could be any size within a tile.”), where the previously determined minimum depth value is compared to the maximum depth value for the primitive to determine visibility, as described in column 51 lines 7-11 (“...comparing the maximum z value for a plurality of stamp

positions with a minimum z value of the current primitive and setting corresponding stamp selection bits...“) and the visibility is indicated using a selection bit or occlusion flag as described in column 37 lines 6-8 (“...indicates is potentially visible (using the row selection bits...“). It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al, Larson and Duluk, Jr. et al. because this combination would provide the determination of visible pixels ot be rendered thereby reducing processing time used to render occluded pixels.

Regarding claim 31, Olsen et al. teaches processing a pixel for the graphics primitive, as described in column 4 lines 39-41 (“...the circuit 33 typically compares the incoming z pixel data to the current z pixel data using a function that is true if the incoming z pixel data is in front of the current z pixel data...“) responsive to the occlusion flag comprises rendering the pixel without comparing a depth value thereof to an occlusion threshold value responsive to detecting that the occlusion flag indicates that the graphics primitive is not occluded and is visible within the tile in step 47 of Figure 4 where it is shown that if is determined by an occlusion flag or register that the pixel is visible, in which case the flag is set to true as described in column 5 lines 52-56 (“If the value stored in the register 35 has a logical value of 1, then this indicates that at least one pixel from the primitives defining the bounded volume passes the depth comparison function. Thus, at step 48, all the primitives enclosed by the bounded volume are rendered.“), then the process proceeds to steps 48 and 49 to render the pixel and does not return to steps 45-46 to further compare the depth value.

Regarding claim 35, Olsen et al. teaches in column 6 lines 8-13, setting an occlusion flag, or depth compare result register, for respective pixels. Olsen and Wood et al. fail to teach

occlusion flags responsive to comparisons of the maximum depth value to a minimum depth value in the occlusion information cache. Larson teaches establishing a tile occlusion information cache in column 5 lines 63-66, and comparing the maximum depth value with the cached occlusion threshold depth value in column 2 lines 46-51. However, Olsen et al., Wood et al. and Larson fail to teach comparing the maximum depth value to a minimum depth value. Duluk, Jr. et al. teaches comparing the maximum depth value to the minimum depth value in column 51 lines 7-11 (“...comparing the maximum z value for a plurality of stamp positions with a minimum z value of the current primitive and setting corresponding stamp selection bits...”). The motivation to combine the teachings of Olsen et al., Wood et al., Larson and Duluk, Jr. et al. is equivalent to the motivation of claim 30.

#### ***Response to Arguments***

Applicant's arguments filed 2/23/2006 have been fully considered but they are not persuasive.

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The applicant argues that the reference Olsen et al. does not teach an occlusion flag for a tile and processing pixels in rows of pixels in a row-by row manner responsive to the occlusion flags. The examiner maintains the rejection because Olsen et al. teaches processing pixels in rows of pixels in a row-by-row manner

responsive to the occlusion flags in column 5 lines 22-23 where it is described that the depth values of each pixel of the graphics primitive, which contains rows of tiles, are processed, therefore each row would be processed as each pixel comprising the rows would be processed. However, it is stated above in the rejection of claims 1, 15, 23 and 26 that Olsen et al. does not teach defining a plurality of rows of tiles, as described on page 3 line 10: "Olsen et al. fails to teach defining a plurality of rows of tiles", and that the reference Wood et al. was combined with Olsen et al. in the 35 U.S.C. 103(a) rejection of claims 1, 23 and 26, as well as combined in the 35 U.S.C. 103(a) rejection of claim 15, to teach defining tiles comprising rows of pixels in column 1 lines 6-10 and also teaches processing each row of pixels in column 5 lines 15-23 ("...reading the tile fragments in zig-zag raster scanning order, starting at top left, scanning even numbered rows from left to right, odd numbered rows from right to left..."). Therefore, it would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al. with Wood et al. because this combination would provide an accurate determination of occluded pixels within tiles in a graphics display thereby reducing the computation time by providing only the visible pixels to be rendered, as described by Wood et al. in column 9 lines 59-62 ("The present architecture ignores fully occluded fragments and only textures pixel fragments that are visible, reducing traffic from the texture memory...").

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The applicant argues that the reference Olsen et al. used in the 35 U.S.C. 103(a) rejection of claims 1, 15, 23 and 26 does not

teach setting occlusion flags for respective tiles of a row of tiles. The examiner maintains the rejection because Olsen et al. teaches setting the occlusion flag which indicates non-occlusion in column 5 lines 42-45, where it is described that the depth compare result register is set to a value of 1 to indicate non-occlusion of the pixels and Wood et al. teaches processing each row of pixels in column 5 lines 15-23 (“...reading the tile fragments in zig-zag raster scanning order, starting at top left, scanning even numbered rows from left to right, odd numbered rows from right to left...”) and also describes defining tiles comprising rows of pixels in column 1 lines 6-10. Therefore, it would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al. with Wood et al. because this combination would provide a reduced computation time by determining the occluded pixels and provide only the visible pixels to be rendered, as described by Wood et al. in column 9 lines 59-62 (“The present architecture ignores fully occluded fragments and only textures pixel fragments that are visible, reducing traffic from the texture memory...”).

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The applicant argues that the reference Larson used in the 35 U.S.C. 103(a) rejection of claim 4 does not teach determining a maximum depth value for the graphics primitive. The examiner maintains the rejection because Larson teaches determining a maximum depth value for the graphics primitive in column 1 lines 59-60 (“The maximum Z value corresponds to the largest Z value of a region of Z values.”), where the z value is the maximum depth value, as known in the art and described in column 1

lines 19-25 ("Z buffer depth comparison tests are used to determine whether a new Z coordinate received in the frame buffer controller corresponds to a pixel that will be visible when displayed..."). It would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide decreased processing time of pixel of the graphics primitive during determination of occlusion by storing occlusion flags in a cache memory, as described by Larson in column 5 lines 28-31 ("...cache memory is used to reduce the number of Z buffer accesses and Z limit buffer accesses and to speed up the Z buffer depth comparison test.").

In response to applicant's arguments against the references individually, one cannot show nonobviousness by attacking references individually where the rejections are based on combinations of references. See *In re Keller*, 642 F.2d 413, 208 USPQ 871 (CCPA 1981); *In re Merck & Co.*, 800 F.2d 1091, 231 USPQ 375 (Fed. Cir. 1986). The applicant argues that the reference Larson used in the 35 U.S.C. 103(a) rejection of claim 18 does not teach a maximum depth value and whether a maximum depth for the graphics primitive in the tile is less than a previously established minimum non-occlusion depth value for the tile. The examiner maintains the rejection because Larson teaches determining a maximum depth value for the graphics primitive in column 1 lines 59-60 ("The maximum Z value corresponds to the largest Z value of a region of Z values."), where the z.value is the maximum depth value, as known in the art and described in column 1 lines 19-25 ("Z buffer depth comparison tests are used to determine whether a new Z coordinate received in the frame buffer controller corresponds to a pixel that will be visible when displayed..."). Larson also teaches that a depth value, or z value, is compared to determine if that value is less than an established non-occluded, or visible,

minimum depth value in column 5 lines 19-21 ("...the Z values for the incoming pixels of a particular region will be either greater than the Z MAX value or less than the Z MIN value."), therefore the received value could be a maximum z value in which that received value is compared with the minimum z value, as described in column 5 lines 7-11 ("If the current pixel Z value is not greater than or equal to the Z value read out of the Z buffer, then the pixel is visible and the current pixel Z value must be written to the Z buffer and the Z MIN and Z MAX values must be updated...") and in column 6 lines 23-30 ("The received Z value is then compared with the Z MIN value. If the received Z value is less than Z MIN, the primitive is visible."), in which case the Z MIN value is the minimum non-occluded or visible depth value. Therefore, it would have been obvious to one of ordinary skill in the art to combine the teachings of Olsen et al., Wood et al. and Larson because this combination would provide an apparatus that determines occlusion of each row of pixels comprising the tiles for a graphics primitive, which would reduce computation time by providing only the visible pixels to be rendered, as described by Wood et al. in column 9 lines 59-62 ("The present architecture ignores fully occluded fragments and only textures pixel fragments that are visible, reducing traffic from the texture memory...").

### *Conclusion*

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Said Broome whose telephone number is (571)272-2931. The examiner can normally be reached on 8:30am-5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ulka Chauhan can be reached on (571)272-7782. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

S. Broome  
5/8/06 *SB*

*Ulka Chauhan*  
ULKA CHAUHAN  
SUPERVISORY PATENT EXAMINER